

(12) UK Patent Application (19) GB (11) 2 256 513 (13) A

(43) Date of A publication 09.12.1992

(21) Application No 9210282.1

(22) Date of filing 13.05.1992

(30) Priority data
(31) 03120761

(32) 27.05.1991

(33) JP

(71) Applicants
Hitachi Ltd

(Incorporated in Japan)

6 Kanda Surugadai 4-chome, Chiyoda-ku, Tokyo,
Japan

Hitachi Computer Engineering Co Ltd

(Incorporated in Japan)

1 Horiyamashita, Hadano-shi, Kanagawa-ken,
Japan

(72) Inventors
Toyohisa Imada
Yasusuke Takeshima

(51) INT CL⁵
G06F 12/02

(52) UK CL (Edition K)
G4A AFN

(56) Documents cited
EP 0473802 A1 EP 0247374 A2

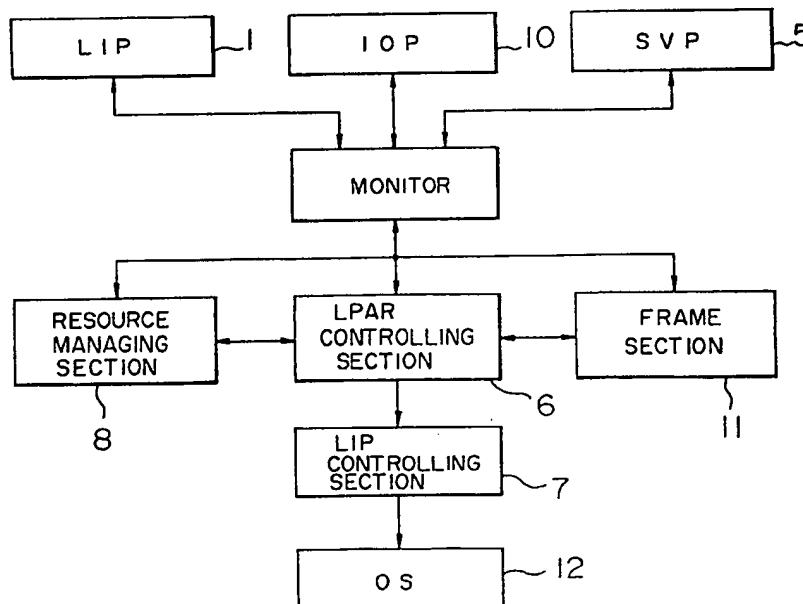
(58) Field of search
UK CL (Edition K) G4A AFN AMX ANV
INT CL⁵ G06F 12/02 12/06

(74) Agent and/or Address for Service
Mewburn Ellis
2 Cursitor Street, London, EC4A 1BQ, United Kingdom

(54) Dynamically relocating main storage to virtual machine

(57) A virtual machine system having a plurality of virtual machines, in which each virtual machine can be relocated to a new storage area of main storage without interfering with any other virtual machine. The relocation is carried out by move instructions issued by a service processor 5 upon receipt of a relocate command. These instructions pass control to a controlling section 6 which instructs a logical instruction processor (LIP) controlling section 7 to temporarily stop the LIP of the particular virtual machine. Next a check is made by the resource managing section 8 to determine whether the virtual machine can be moved. Once that has been checked, the virtual machine is relocated on the main storage area in accordance with a designated address by the relocate command and the controlling section 7 restores operation of the virtual machine.

FIG. 1B



GB 2 256 513 A

FIG. 1A

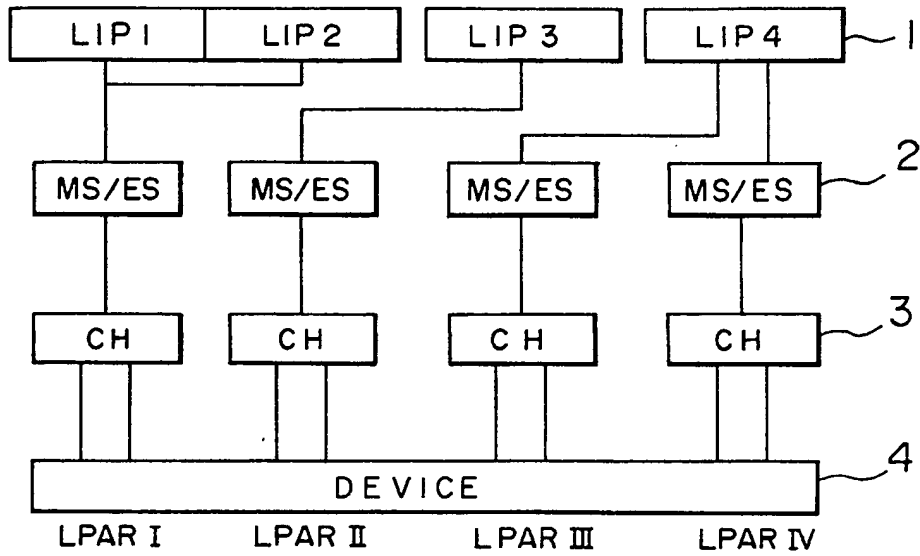


FIG. 1B

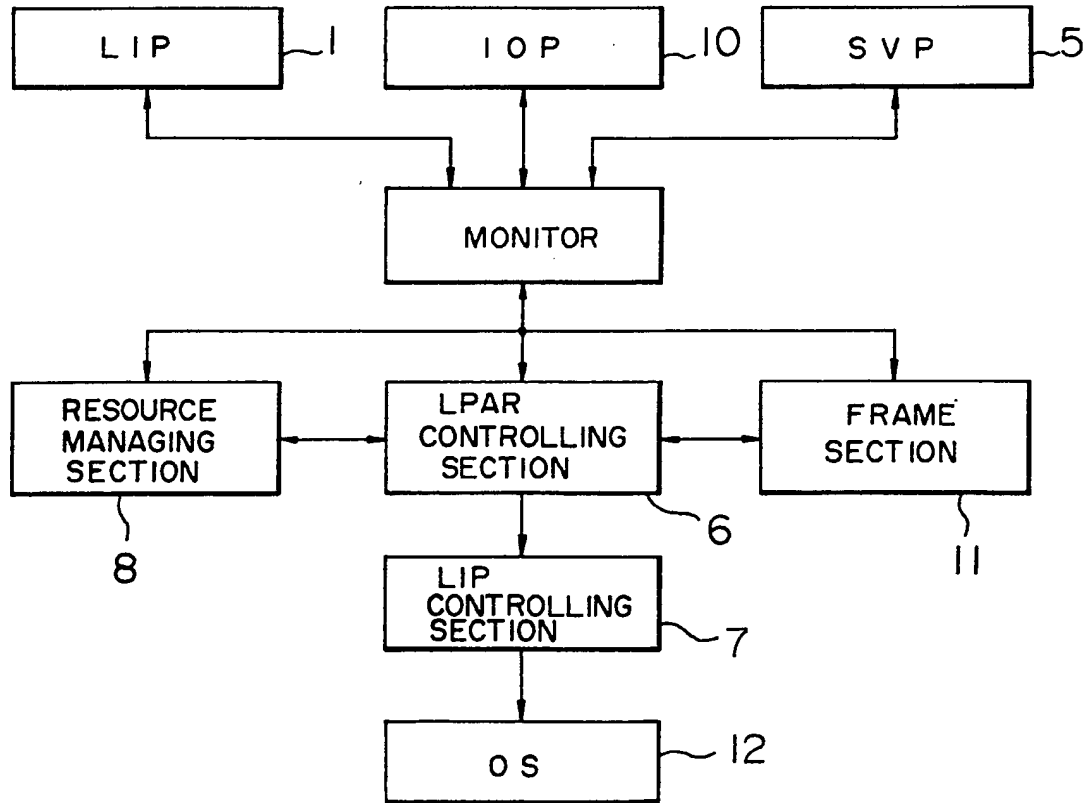


FIG. 2

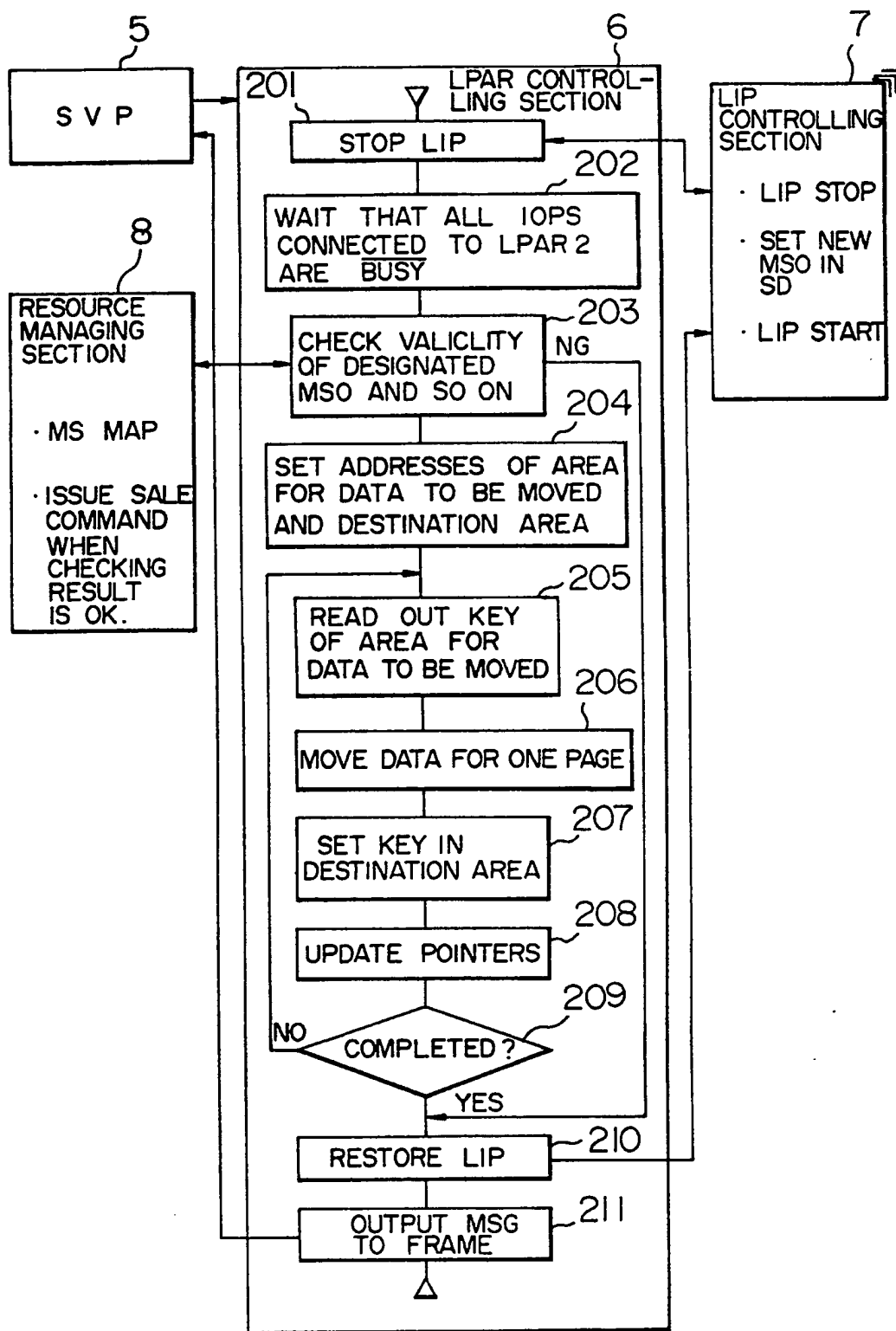


FIG. 3A

LPAR NAME	MS AREA ORIGIN	MS AREA SIZE	M BYTE (MB)
LPAR 1	0	128	0
LPAR 2	128	256	125

AVAILABLE PHYSICAL MAIN STORAGE AREA : 509MB

FIG. 3B

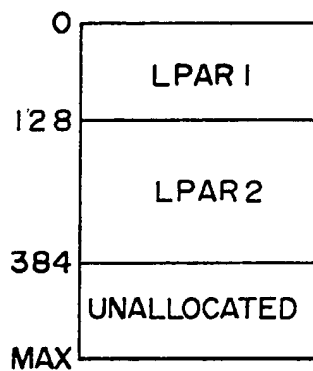
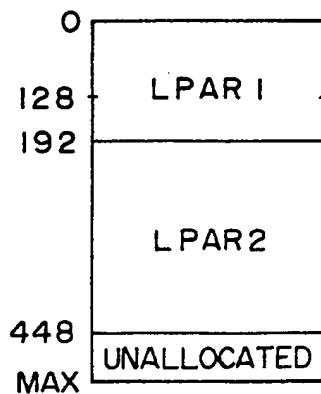


FIG. 4A

LPAR NAME	MS AREA ORIGIN	MS AREA SIZE	M BYTE (MB)
LPAR 1	0	128	64
LPAR 2	192	256	61

AVAILABLE PHYSICAL MAIN STORAGE AREA : 509MB

FIG. 4B



METHOD OF AND SYSTEM FOR DYNAMICALLY
RELOCATING MAIN STORAGE AREA TO VIRTUAL MACHINE

1

The present invention relates to a virtual machine system in which virtual machines are allocated to logically partitioned areas (LPARs), and in particular, to a virtual machine system in which virtual machines can be relocated to a main storage area.

In the conventional art, there has been known a technology for relocating a main storage area to a virtual machine in the virtual machine system in which the main storage area allocated to the virtual machine is expanded or degenerated. However, in the above conventional technology, the origin of the main storage area assigned to the virtual machine, i.e., a start address of the allocated area is not altered. Consequently, in a case where a central portion of the main storage area is physically allocated to a first virtual machine and the first virtual machine is using a basic software such as an operating system (OS), when there occurs a request to allocate to a second virtual machine a portion of the main storage area, it is impossible in some cases to allocate a successive portion of the main storage area necessary for the second virtual machine. In such a case, according to the conventional art, the first

1 virtual machine is required to be once stopped to re-
establish the origin of the main storage area allocated
thereto and then an initial program loader (IPL) is
effected thereafter.

5

The present invention therefore seeks
to provide a method of and a virtual machine
system for dynamically relocating a main storage area
wherein when the main storage area cannot be allocated to
10 the second virtual machine, as described above, the first
virtual machine is left remained in its state such that
the main storage origin assigned to the first virtual
machine is moved to a specified address to allocate a
resultant free area to the second virtual machine for
15 setting the second virtual machine in an operable state,
thereby improving efficiency and usability of the
information processor.

Therefore, a
relocation method according to the present invention
20 includes the steps of stopping a virtual machine in a
virtual machine system constituted with a plurality of
virtual machines, relocating the virtual machine on a
main storage area in accordance with an address
designated by a relocation command, and restoring the
25 virtual machine based on the designated address.

According to the present invention, when the
plural virtual machines physically use the whole main

1 storage area, if portions of the main storage area
allocated to at least two virtual machines are to be
dynamically exchanged, or alternatively, if a portion of
the main storage area allocated to one of the plural
5 virtual machines are to be expanded while the plural
virtual machines are in operation, portions of the main
storage area allocated to other virtual machines can be
freely moved. For this reason, without restoration of
the plural virtual machines in operation, the origin of
10 the main storage allocated to each virtual machine in
operation, i.e., the head address of the allocated area
can be freely changed, thereby substantially continuing
the operation state of the virtual machine. As a result,
the relocation of the main storage area can be performed
15 for the specified virtual machine almost without exerting
substantially influence upon the user of the virtual
machine, which improves efficiency and usability of the
information processing apparatus.

20 These and other aims and advantages of the
present invention will become apparent by reference to
the following description and accompanying drawings
wherein:

Fig. 1A is a schematic block diagram showing
25 the hardware resource configuration of a virtual machine
system in an embodiment of the present invention;

Fig. 1B is a block diagram showing the

1 architecture of the virtual machine system;

Fig. 2 is an explanatory diagram for explaining the procedure of moving a main storage area assigned to a virtual machine in the virtual machine system of the

5 present invention;

Figs. 3A and 3B are explanatory diagrams for respectively explaining the allocated state of the virtual machine to the main storage area allocation before the movement of the allocated main storage area;

10 and

Figs. 4A and 4B are explanatory diagrams for respectively explaining the relocated state of the virtual machine to the main storage area after the movement of the allocated main storage area.

15

Referring now to the accompanying drawings, description will be given in detail of a virtual machine system according to an embodiment of the present invention in which a virtual machine can be relocated to
20 a new storage area of a main storage during its operation.

First, the hardware resource constitution of the virtual machine system of the present invention will be described with reference to Fig. 1A. The virtual
25 machine system has a new partition configuration of the main storage area called logically partitioned areas (LPARs), and the hardware resources thereof include four

1 logical instruction processors (LIP1 to LIP4) 1, main
storage sections/virtual storage sections (MS/VS) 2,
channel paths (CH) 3, and a device 4. The device 4 is
divisionally used by four virtual machines. As can be
5 seen from this figure, the logical instruction processors
IP1 and IP2 correspond to a virtual machine LPAR I. The
logical instruction processor LIP3 corresponds to a
virtual machine LPAR II. The logical instruction
processor LIP4 corresponds to virtual machines LPAR III
10 and LPAR IV.

Fig. 1B shows the architecture of a virtual
machine system including a plurality of virtual machines.
Each virtual machine includes a logical instruction
processor (LIP) 1. A service processor (SVP) 5 generates
15 a move instruction in response to a relocate command. A
resource managing section 8 manages and controls the
allocated state of resources to each of the plural
virtual machines. In response to a check request
inputted thereto, this section 8 checks to determine
20 whether or not a virtual machine designated by the
request can be relocated on a main storage area. When
the designated virtual machine is relocated by an LPAR
controlling section 6, the relocated area of the main
storage area is notified to all input/output processors
25 10 allocated to the designated virtual machine. An LIP
controlling section 7 is provided for each virtual
machine to control operations of the logical instruction
processor (LIP) 1 of the virtual machine. Namely, the

1 section 7 stops the operation of the LIP 1 in the
designated virtual machine and then restores the machine
in accordance with an address specified by an SIE (start
interpretive execution) command. The LPAR controlling
5 section 6 operates in response to a move instruction.
When the operation of the designated virtual machine is
stopped, the section 6 awaits a state in which all the
I/O processors 10 assigned to the designated virtual
machine is in other than a busy state to generate and
10 output to the resource managing section 8 the check
request when the operation of the designated machine is
stopped. Thereafter, when it is determined by the
managing section 8 that the designated virtual machine
can be relocated on the main storage area, the section 6
15 relocates the designated virtual machine in accordance
with the address designated by the move instruction.

When the relocation instruction relates to
expansion of a main storage area assigned to any other
virtual machine adjacent to the designated virtual
20 machine, the service processor 5 generates an expand
instruction after the designated machine is restored so
as to expand the area to be assigned to the other virtual
machine in response to the expand instruction. When the
relocation instruction relates to exchange of main
25 storage areas respectively allocated to two designated
virtual machines, the controlling section 6 controls the
LIP controlling sections 7 to respectively stop the two
designated virtual machines, reads out a content stored

- 1 in the main storage area allocated to one of the two
designated virtual machines to store the read out content
in the other of the two virtual machines repeats the
reading out and storing operations over the main storage
5 area allocated to the one virtual machine, and restores
these virtual machines.

The hardware resources are allocated to LPARs
as logical virtual machines in the exclusive or time-
sharing fashion in a LPAR mode. That is, in the hardware
10 constitution, a logical instruction processor 1 is
allocated to an LPAR in a time-sharing manner, whereas
the remaining hardware components are allocated to the
LPAR in a exclusive fashion.

- A main storage area allocated to each LPAR,
15 i.e., an LPAR area can be dynamically changed through the
move processing as shown in Fig. 2. That is, the origin,
namely, the head address of each LPAR area can be
arbitrarily changed in the physical main storage area.
The move processing is carried out by the service
20 processor (SVP) 5 having an LPAR frame to provide an
operation interface for each LPAR, the LPAR controlling
section 6 for controlling operation instructions and the
like to each LPAR, the LIP controlling section 7 for
controlling a guest, e.g., the operating system (OS) in
25 response to an instruction from the controlling section 6
and achieving a simulate processing for a command requir-
ing a simulation, and the resource managing section 8 for
managing information of the physical resources of all the

1 LPARs, as shown in Fig. 2.

Subsequently, referring to Fig. 2, description
will be given of an example of the move processing in a
case where two LPARs, i.e., LPAR1 and LPAR2 are operated,
5 for example.

First of all, prior to initiation of the
operation of each LPAR, the main storage area is
logically partitioned to define areas for the LPAR1 and
LPAR2. Fig. 3A shows a main storage (MS) origin and an
10 MS size for each of the defined areas. An MS gap
indicates the size of an unused area between the LPAR1
area and the LPAR2 area which is at a higher address area
in the main storage. In this example, since the MS gap
is absent between the LPAR1 and LPAR2 areas, the MS gap
15 of the LPAR1 is set to "0". Moreover, in the allocated
state of the main storage when two virtual machines LPAR1
and LPAR2 are operating on the two LPAR areas thus
defined, areas from 0 MB to 128 MB and from 129 MB to 384
MB are respectively allocated to the LPAR1 and LPAR2 and
20 the remaining 125 MB area from 385 MB to 509 MB is not
allocated. During the operations of the LPAR1 and LPAR2
under this circumstance, when it is required to expand
the main storage area allocated to the LPAR1, the main
storage moving (MOVE MS) function of the present
25 invention can be utilized.

First, assume that setting of the MS origin to
192 MB for the LPAR2, i.e., MS0 = 192 is specified by a
MOVE MS command (to be represented as a MVSTOR command

1 herebelow) through the LPAR frame on the SVP 5, for
example. In response to this command, control is passed
from the SVP5 to the LPAR controlling section 6 for the
LPAR2. In the controlling section 6, an instruction is
5 sent to the LIP controlling section 7 to temporarily stop
or suspend an LIP of the LPAR2. In response thereto, the
LIP controlling section 7 sets the LIP, namely, the guest
of the LPAR2, i.e., the operating system to the stop
state (step 201).

10 There is achieved a wait operation to await for
at most ten seconds that where all the I/O processors
linked with the LPAR2 become to be not in a busy state
(step 202). Usually, when the LIP is stopped, the I/O
processor or device can immediately go out of the busy
15 state; however, a ten-second timer is prepared on
assumption of a special case. In a case of a time-out, a
message thereof is outputted to notify any I/O devices in
the busy state. The step 202 is necessitated to prevent
the following possibility. Even when the LIP is stopped
20 in the step 201, the I/O processor is still in operation
and hence the area before movement may possibly be
accessed by the I/O processor in response to an I/O
command.

 A check is made by the resource managing
25 section 8 to determine whether or not the MSO (192 MB in
this case) specified by the command is adequate, namely,
whether or not the LPAR2 area can be moved (step 203).
In short, the resource managing section 8 has the MS maps

1 of all LPARs and determines based on the specified MSO,
whether or not an area from the designated MSO for the
size of the LPAR2 area is used by any other LPAR. In
this embodiment, the area from 192 MB at the MSO to 384
5 MB is used by the LPAR2 and the area beginning from 385
MB is not allocated and hence the LPAR2 area can be
moved. When it is determined that the LPAR2 area can be
moved, the resource managing section 8 sends an SALE (set
address limit extends) command to the I/O processors to
10 notify the LPAR2 area after movement.

Next, the actual memory transfer processing
will be described according to the steps 204 to 209.
First, a start address of the transfer source area, LPAR2
area, and a start address of a transfer destination area
15 are set (step 204). In this embodiment, the start
addresses of the source area and the transfer destination
area are expressed respectively as 384 MB and 448 MB in a
page unit. Namely, the transfer operation is commenced
from the respective start addresses such that the
20 addresses are respectively decremented by one page for
each operation. A key for protecting each page data of
the source area is read out (step 205) to be saved in a
register in the LPAR controlling section 6. Then, the
page data is transferred to the transfer destination area
25 beginning from the address obtained in the step 204 (step
206) so as to set the beforehand obtained key in the
transfer destination area at a predetermined address
(step 207). With this operation, data transfer for one

1 page has been completed. Subsequently, the address of
the source area and the address of the transfer
destination area are then updated (step 208). In this
embodiment, the page count is decremented by one for each
5 transfer to repeatedly conduct the transfer operation for
the MS size of the main storage area allocated to LPAR2
in page units. With the operation above, the MS transfer
operation is finished (step 209).

Finally, in order to restore the LIP, an
10 instruction is issued to the LIP controlling section 7 in
response to a start interpretive execution (SIE) command.
In response thereto, the LIP controlling section 7 sets a
new MSO provided in the parameters of the SIE command and
then restores the LIP. Namely, the guest of the LPAR2 is
15 started (step 210). Moreover, to the frame on the SVP5,
a result of the MVSTOR command is reported (step 211).
In this embodiment, the MVSTOR command is normally
terminated. In this case, after the termination of the
MVSTOR command, the allocated state of the main storage
20 to the LPAR1 and LPAR2 are as shown in Fig. 4A. Since an
MS gap is produced for the LPAR1 by the MVSTOR command,
the LPAR1 area can be increased by use of a conventional
command as much as there is the MS gap. Furthermore, the
area allocation states of the LPAR1 and LPAR2 become to
25 be as shown in Fig. 4B, namely, the LPAR2 area can also
be expanded toward the upper-address direction of the
main storage by 61 MB.

As above, in accordance with the present

1 invention, the main storage allocated to the LPAR1 can be
expanded almost without exerting any influence upon the
operating state of the LPAR2. As can be understood from
the description above, according to the virtual machine
5 system of the present invention, it is possible to
provide an MVSTOR command for arbitrarily changing the
origin of each LPAR area in the physical main storage in
a virtual machine system having the LPAR mode function.
This consequently unnecessitates the conventional
10 operation in which the virtual machine in operation is
stopped to set the main storage origin in another main
storage area and to re-initiate the IPL after the
relocation of the LPAR area, resulting in enabling the
virtual machine to be continuously operated.

15 That is, the virtual machine system of the
present invention includes an information processing
apparatus, a service processor providing an operator
interface of the information processing apparatus, and a
hardware mechanism for controlling a plurality of virtual
20 machines. The hardware mechanism is logically parti-
tioned in an exclusive manner or in a time-sharing manner
and the origin of a main storage area of each of the
plural virtual machines is altered to move the allocated
main storage area.

25 Moreover, when the plural virtual machines use
the whole area of the physical main storage, it is
possible to dynamically exchange main storage areas
allocated to at least two virtual machines.

1 Furthermore, when the main storage area of one
of the plural virtual machines is to be expanded while
the plural virtual machines are in operation, the main
storage areas of other virtual machines can be freely
5 moved. In other words, when the address of the main
storage area allocated to a specified virtual machine is
supplied, the operation is attained by the service
processor so as to notify the alteration of the main
storage area origin to the controlling section for the
10 specified virtual machine. The controlling section stops
the logical instruction processor controlling section.
Next, adequacy of a specified main storage area origin is
checked by the resource managing section managing
physical resources of all virtual machines. When it is
15 confirmed that the area can be moved to the specified
area, the controlling section for the virtual machine
moves data in the page unit. A range of the new main
storage area of the virtual machine is then notified by
the SALE command to input/output processors coupled to
20 the virtual machine and the main storage area origin is
finally varied in the parameters of the SIE command,
thereby restoring the instruction processor controlling
section. Resultantly, the main storage area of the
specified virtual machine can be transferred almost
25 without exerting any influence upon the user operating
the virtual machine.

Description has been specifically given of the
invention according to an embodiment. However, the

1 present invention is not restricted by the embodiment and can be naturally modified and varied in various manners without departing from the spirit and scope of the invention.

CLAIMS

1. A method of operating a virtual machine system including a plurality of virtual machines, the method being for relocating on a main storage area a virtual machine specified by a relocate command, the method comprising the steps of:
 - temporarily stopping a virtual machine;
 - relocating said virtual machine on the main storage area in accordance with an address designated by the relocate command; and
 - restoring said virtual machine based on the relocated main storage area.
2. A method according to Claim 1, wherein said temporarily stopping step comprises the steps of:
 - temporarily stopping a logical instruction processor in said virtual machine; and
 - waiting until all input/output processors assigned to said logical instruction processor are set to a state other than a busy state.
3. A method according to Claim 1, wherein said relocating step comprises the steps of:
 - determining in accordance with the designated address, whether or not said virtual machine can be relocated on the main storage area; and
 - relocating said virtual machine in accordance with the address designated by the relocation command when it is determined that said virtual machine can be relocated.
4. A method according to Claim 3, wherein said

relocating step includes the steps of transferring stored contents of the main storage area currently allocated to said virtual machine to a new main storage area determined based on the designated address in the page units.

5. A method according to Claim 1, wherein said restoring step comprises the steps of:

generating a restore command of said virtual machine;

notifying, in response to the instruction, the relocated main storage area to all the input/output processors allocated to said virtual machine; and

restoring said virtual machine based on the designated address.

6. A method according to Claim 1, further comprising the steps of expanding, when the relocate command relates to expansion of a main storage area allocated to another virtual machine adjacent to the virtual machine, the main storage allocated to the other virtual machine after said restoring step.

7. A method according to Claim 1, wherein:

the relocate command relates to exchange of main storage areas respectively allocated to two virtual machines designated by the relocate command,

said temporarily stopping step comprises the step of temporarily stopping each of the two virtual machines,

said relocating step comprises the steps of:
reading out contents of corresponding areas in

the main storage areas allocated to the two virtual machines;

storing contents read out from the main storage area allocated to one of the two virtual machines in the corresponding area of the main storage area allocated to the other thereof; and

repeatedly achieving said reading-out step and said storing step for the main storage areas allocated to the two virtual machines, and

said restoring step comprises the step of restoring the two virtual machines.

8. A virtual machine system including a plurality of virtual machines allocated on a main storage, each said virtual machine including a logical instruction processor (LIP), comprising:

a service processor for generating a move command in response to a relocate instruction;

resource managing means for managing allocation states of resource to the virtual machines and controlling the resources;

LIP controlling means provided for each of said virtual machines, for controlling an operation of the logical instruction processor of said each virtual machine; and

virtual machine controlling means responsive to the move command, for relocating a virtual machine designated by the relocate command on the main storage area in accordance with an address designated in the move

command when an operation of said designated virtual machine is temporarily stopped, and for controlling said LIP controlling means corresponding to said designated virtual machine to restore said designated virtual machine based on the designated address.

9. A virtual machine system according to Claim 8, wherein said virtual machine controlling means further comprises means for controlling said corresponding LIP controlling means to temporarily stop said logical instruction processor of said designated virtual machine, awaiting that all input/output processors assigned to said designated virtual machine are set to a state other than a busy state.

10. A system according to Claim 8, wherein said resource managing means comprises means responsive to a check request inputted thereto, for determining in accordance with the designated addresses, whether or not said designated virtual machine can be relocated on the main storage area, and

said virtual machine controlling means comprises means for generating the check request to output the same to said resource managing means when operation of said designated virtual machine is temporarily stopped and for relocating said designated virtual machine in accordance with the designated address when it is determined by said resource managing means that said designated virtual machine can be relocated.

11. A system according to Claim 8, wherein said

resource managing means further comprises means for notifying, when said designated virtual machine has been relocated by said virtual machine controlling means, a relocation destination to all input/output processors assigned to said designated virtual machine.

12. A system according to Claim 8, wherein said virtual machine controlling means comprises means for transferring contents of the main storage area currently allocated to said designated virtual machine to a new main storage area as a destination of the relocation in units of pages.

13. A system according to Claim 8, wherein:
said service processor comprises means for generating, when the relocate command relates to expansion of a main storage area allocated to one of the virtual machines adjacent to said designated virtual machine, an expand command after said designated virtual machine is restored; and

said virtual machine system further comprises means for expanding, in response to the expand command, the main storage area allocated to said other virtual machine.

14. A system according to Claim 8, wherein:
the relocate command relates to an exchange of main storage areas respectively assigned to two designated virtual machines,

said virtual machine controlling means
comprises means for controlling LIP controlling means

associated with said two virtual machine to stop each of said two virtual machines, for reading out stored contents into an associated area in the main storage areas allocated to the two virtual machines, for storing the read out content of one of the two virtual machines in an associated area of the main storage area allocated to the other thereof, for repeatedly achieving the reading-out operation and the storing operation for the main storage areas allocated to the two virtual machines, and for restoring said two virtual machines.

15. A method of operating a virtual machine system substantially as any one described herein with reference to the accompanying drawings.

16. A virtual machine system substantially as herein described with reference to and as illustrated in the accompanying drawings.

Patents Act 1977
 Examiner's report to the Comptroller under
 Section 17 (The Search Report)

Application number

9210282.1

Relevant Technical fields

(i) UK Cl (Edition K) G4A (AFN, ANV, AMX)

(ii) Int Cl (Edition 5) G06F 12/02, 12/06

Search Examiner

MISS A C CLARKE

Databases (see over)

(i) UK Patent Office

(ii)

Date of Search

30.7.92

Documents considered relevant following a search in respect of claims

1-14

Category (see over)	Identity of document and relevant passages	Relevant to claim(s)
A	EP 0473802 A1 (IBM) whole document	1 at least
A	EP 0247374 A2 (IBM) whole document	1 at least

Category	Identity of document and relevant passages	Relevant to claim(s)

Categories of documents

X: Document indicating lack of novelty or of inventive step.

Y: Document indicating lack of inventive step if combined with one or more other documents of the same category.

A: Document indicating technological background and/or state of the art.

P: Document published on or after the declared priority date but before the filing date of the present application.

E: Patent document published on or after, but with priority date earlier than, the filing date of the present application.

&: Member of the same patent family, corresponding document.

Databases: The UK Patent Office database comprises classified collections of GB, EP, WO and US patent specifications as outlined periodically in the Official Journal (Patents). The on-line databases considered for search are also listed periodically in the Official Journal (Patents).